AMENDMENT TO THE CLAIMS

Please accept the following amended claims. Claims 1, 10, 11 and 20 have been amended. Claims 2 and 12 were previously canceled. (All claims listed)

1. (Currently amended) A method of establishing thread priority in a single processor comprising:

assigning a value in memory to indicate which of a plurality of threads executed by said single processor has a higher priority; ~~and~~

allocating a resource between said plurality of threads depending on a priority assigned to each thread~~.~~; and

providing a counter with a predetermined value for said plurality of threads depending on the priority assigned to each thread.

2. (Canceled)

3. (Previously presented) The method of claim 1 wherein in said allocating step, a first thread is given greater access to the resource than other threads when said first thread is assigned a higher priority than said other threads.

4. (Previously presented) The method of claim 1, wherein in said allocating step, the other threads are given greater access to the resource than the first thread when said first thread is assigned a higher priority than the other threads and is not using said resource.

5. (Original) The method of claim 3 wherein said resource is a unit in a processor system.

6. (Original) The method of claim 5 wherein said resource is a decode unit.

7. (Original) The method of claim 6 further comprising:

providing instructions from a first thread to a first queue;

providing instructions from a second thread to a second queue;

supplying a first number of instructions to said decode unit from said first queue;

supplying a second number of instructions to said decode unit from said second queue;

selecting said first and second numbers based on said value in memory.

8. (Original) The method of claim 3 wherein said resource is a bus.

9. (Original) The method of claim 8 further comprising:

providing bus requests from a first thread to a first queue;

providing bus requests from a second thread to a second queue;

servicing a first number of bus requests from the first queue;

servicing a second number of bus requests from said second queue; and

selecting said first and second numbers based on said value in memory.

10. (Currently amended) A method of establishing thread priority in a single processor comprising:

assigning a value in an APIC TPR register for a thread via execution of operating system code to indicate which of a plurality of threads executed by said single processor has a higher priority; ~~and~~

allocating a resource between said plurality of threads depending on a priority assigned to each thread~~.~~; and

providing a counter with a predetermined value for said plurality of threads depending on the priority assigned to each thread.

11. (Currently amended) An apparatus for establishing thread priority in a single processor comprising:

a memory to store a value to indicate which of a plurality of threads to be executed by said single processor has a higher priority; ~~and~~

a resource allocated between said plurality of threads depending on a priority assigned to each thread in said memory~~.~~; and

a counter loaded with a predetermined value for each thread in said memory depending on the priority assigned.

12. (Canceled)

13. (Previously presented) The apparatus of claim 11 wherein a first thread is given greater access to the resource than other threads when said first thread is assigned a higher priority than said other threads.

14. (Previously presented) The apparatus of claim 11 wherein the other threads are given greater access to the resource than the first thread when said first thread is assigned a higher priority than the other threads and is not using said resource.

15. (Original) The apparatus of claim 13 wherein said resource is a unit in a processor system.

16. (Original) The apparatus of claim 15 wherein said resource is a decode unit.

17. (Original) The apparatus of claim 16 further comprising:

a first queue to store instructions from a first thread;

a second queue to store instructions from a second thread;

control logic coupled to said first and second queues and said decode unit, said control logic to permit a first number of instructions to be transferred to said decode unit then a second number of instructions to be transferred to said decode unit, said first and second numbers being selected based on said value in memory.

18. (Original) The apparatus of claim 13 wherein said resource is a bus.

19. (Original) The apparatus of claim 18 further comprising:

a bus unit including

a first queue storing bus requests from a first thread;

a second queue storing bus requests from a second thread;

control logic coupled to said first and second queues, said control logic to control servicing of a first number of bus requests from the first queue and a second number of bus requests from said second queue, said first and second number being selected based on said value in memory.

20. (Currently Amended) An apparatus for establishing thread priority in a single processor comprising:

an APIC TPR register for a thread wherein execution of operating system code causes a value to be stored in said register to indicate which of a plurality of threads to be executed by said single processor has a higher priority; ~~and~~

a resource allocated between said plurality of threads depending on a priority assigned to each thread in said memory~~.~~; and

a counter loaded with a predetermined value for each thread in said memory depending on the priority assigned.